The University of Central Punjab

Faculty of Information Technology

Data Structures and Algorithms
Fall 2022

Lab 02	
Topic	• Understanding Classes • Working on Three Different files • Abstract Classes • Templates • Arrays
Objective	The basic purpose of this lab is to revise some preliminary concepts of C++ that have been covered in the course of Introduction to Computing and Programming Fundamentals and Object-Oriented Programming.

Instructions:

- Indent your code.
- Comment your code.
- Use meaningful variable names.
- Plan your code carefully on a piece of paper before you implement it.
- Name of the program should be the same as the task name. i.e. the first program should be Task_1.cpp
- **void main() is not allowed. Use int main()**
- **You have to work in multiple files. i.e separate .h and .cpp files**
- **You are not allowed to use system("pause")**
- **You are not allowed to use any built-in functions**
- **You are required to follow the naming conventions as follow:**
    - **Variables:** firstName; (no underscores allowed)
    - **Function:** getName(); (no underscores allowed)
    - **ClassName:** BankAccount (no underscores allowed)

Students are required to complete the following tasks in lab timings.

Task 1

Create a C++ generic abstract class named as **List**, with the following:

Attributes:

1. Type * arr;
2. int maxSize;
3. int currentSize;

Functions:

virtual void addElementAtFirstIndex(Type) = 0;

- Should add the element at the first position of the **List**

virtual void addElementAtLastIndex(Type) = 0;

- Should add the element at the last position of the **List**

virtual Type removeElementFromEnd() = 0;

- Should remove the element from the last position of the **List**

virtual void removeElementFromStart() = 0;

- Should remove the element from the first position of the **List**

virtual void updateElementsFromList() = 0;

- Should Update element of given position of the **List**

- Write parameterized constructor with default arguments for the above class.
- Write Destructor for the above class.

Task 2

Create a menu based program for the following functions, using the class made in task 1, make a class named as **MyList**, having following additional functionalities:

bool empty() : Returns whether the MyList is empty or not

bool full() : Returns whether the MyList is full or not
int size() : Returns the current size of the MyList

bool insertAt(int index, T value): Adds a value at the index passed to the function, returns true if the index is present and value is added else returns false.
Type last() : Returns the last element of the MyList

bool search(Type): Returns true if the searched value is present in the list else returns false

- Write parameterized constructor with default arguments for the above class..
- Write Destructor for the above class.

Task 3

Write a function, Given an array and a number k where k is smaller than the size of the array, we need to find the k'th smallest element in the given array. It is given that all array elements are distinct.

***Input**: arr[] = {1, 15, 4, 3, 20, 30}*

k = 3

***Output**: 4*

***Input**: arr[] = {1, 15, 4, 3, 20, 30}*

k = 4

***Output**: 15*

Task 4 (Class Participation Task)

Compute the complexity of the following codes, according to their line by line execution. Finally compute their big(O) complexity.

a.

```
void phytagorean(int value)
{
  for(int i = 1; i <= value; i++)
  {
        for(int j = 1; j <= value; j++)
        {
              for(int k = 1; k <= value; k++)
              {
              int num1 = (i*i) + (j*j);
              int num2 = (k*k);
              if (num1 == num2)
                    cout<<"Pair is: (" << i << ", " << j << ", " << k << ")" << endl;
              }
```

```
                }
        }
    }
```

b.

```
    void RangeCheck(int arr[],int sixe, int num1, int num2){
        int counter = 0;
        for (int i=0; i<sixe; i++)
        {
            if (arr[i] >= num1 && arr[i] <= num2)
            {
                counter++;
                cout << arr[i] << "is in the range " << endl;
            }
        }
        cout << "Total Numbers in range are: " << counter << endl;
    }
```